Available online at www.sciencedirect.com
ScienceDirect
International Journal of Approximate Reasoning
44 (2007) 224–243
INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

Hybrid possibilistic networks

Salem Benferhat *, Salma Smaoui

CRIL – CNRS 2499, Université d'Artois, Rue Jean Souvraz SP18, 62307 Lens Cedex, France

Received 15 December 2005; received in revised form 30 June 2006; accepted 31 July 2006
Available online 25 September 2006

Abstract

Possibilistic networks and possibilistic logic are two standard frameworks of interest for representing uncertain pieces of knowledge. Possibilistic networks exhibit relationships between variables while possibilistic logic ranks logical formulas according to their level of certainty. For multiply connected networks, it is well-known that the inference process is a hard problem. This paper studies a new representation of possibilistic networks called hybrid possibilistic networks. It results from combining the two semantically equivalent types of standard representation. We first present a propagation algorithm through hybrid possibilistic networks. This inference algorithm on hybrid networks is strictly more efficient (and confirmed by experimental studies) than the one of standard propagation algorithm.

Keywords: Possibilistic logic; Possibilistic networks; Possibility theory

1. Introduction

Possibilistic and probabilistic networks [1–4] are important tools proposed for an efficient representation and analysis of uncertain information.

Their success is due to their simplicity and their capacity of representing and handling independence relationships which are important for an efficient management of uncertain pieces of information.

* Corresponding author. Tel.: +33 3 21 79 17 79; fax: +33 3 21 79 17 70.
E-mail addresses: benferhat@cril.univ-artois.fr (S. Benferhat), smaoui@cril.univ-artois.fr (S. Smaoui).

0888-613X/$ - see front matter
doi:10.1016/j.ijar.2006.07.012

Possibilistic networks are directed acyclic graphs (DAG), where each node encodes a variable and every edge represents a "causal" or an "influence" relationship between two variables. Uncertainty is expressed by means of conditional possibility distributions for each node in the context of its parents.

The inference in possibilistic graphs depends on the structure of a DAG. For simply connected graphs, the inference process can be achieved in a polynomial time. However, for multiply connected graphs, the propagation algorithm is expensive. It generally requires a graphical transformation from the initial graph to another tree structure such as a junction tree, where nodes in this tree are sets of variables called clusters. The propagation algorithm efficiency depends on clusters' size, and the space complexity is a function of cartesian product of cluster variables' domains.

Another standard way to represent uncertain information in possibility theory framework is possibilistic logic. Possibilistic logic is an extension of classical logic. A weight is associated with each propositional formula. This weight represents the formula's priority regarding other formulas. The set of such weighted formulas is called possibilistic knowledge base.

This paper proposes a new representation of uncertain information in possibilistic networks called hybrid possibilistic networks. The idea is to continue the use of graphical structure to represent independence relations, and to use possibilistic logic to locally represent the uncertainty associated with each node and its parents. Namely, in our representation local uncertainty is no longer represented by conditional possibility distributions but by possibilistic knowledge bases.

The main advantage of this representation concerns space complexity. For instance, in singly connected networks, it may happen that for a given variable the number of parents can be very high. In this case, it may be impossible to provide the conditional possibility distributions for this variable. In our framework, one can only provide a compact representation of these conditional possibility distributions by means of possibilistic knowledge bases. A similar remark also holds for multiply connected networks. Namely, during the junction tree construction, it may happen that the size of clusters can be very large. In this case, it can be impossible to construct possibility distributions associated with clusters. Our representation enables us to represent possibilistic knowledge bases associated with large clusters.

This hybrid representation generalizes the two well-known representation frameworks: possibilistic logic and possibilistic networks.

This paper is an extended and revised version of the conference papers [5,6]. It is organized as follows:

Section 2 fixes the notations and presents a brief background on standard possibilistic frameworks: possibilistic logic and possibilistic networks. Section 3 introduces our hybrid representation of possibilistic networks. The adaptation of propagation algorithms for multiply connected networks is presented in Section 4. Experimental results are presented in Section 6.

2. Possibilistic logic and possibilistic networks

2.1. Notations

Let $V = \{A_1, A_2, \ldots, A_n\}$ be a set of variables. D_{A_i} denotes the finite domain associated with the variable A_i. For the sake of simplicity, and without lost of generality, variables considered here are assumed to be binary. a_i denotes any of the two instances of A_i and $\neg a_i$ represents the other instance of A_i. $\varphi, \psi, \ldots$ denote propositional formulas (called also events) obtained from V and logical connectors $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (propositional negation). $\top$ and $\bot$, respectively, denote tautologies and contradictions.

$\Omega = \times_{A_i \in V} D_{A_i}$ represents the universe of discourse and ω, an element of Ω, is called an *interpretation*. It is either denoted by tuples $(a_1, \ldots, a_n)$ or by conjunctions $(a_1 \wedge \cdots \wedge a_n)$, where a_i's are respectively instance of A_i's. In the following, $\models$ denotes the propositional logic satisfaction. $\omega \models \varphi$ means that ω is a model of φ.

2.2. Possibility measure and possibility distribution

Let us introduce the basic notions concerning possibility theory [7]. Uncertainty on an event φ can be described by two dual measures: possibility measure Π and necessity measure N.

The possibility measure Π is a function that associates to each formula φ a weight in a unit interval $[0, 1]$. Π should satisfy the following requirements:

(i) $\Pi(\top) = 1$
(ii) $\Pi(\bot) = 0$
(iii) $\Pi(\varphi \vee \psi) = \max(\Pi(\varphi), \Pi(\psi))$, and
(iv) if φ and ψ are logically equivalent (namely they have the same set of models) then $\Pi(\varphi) = \Pi(\psi)$. $\Pi(\varphi)$ represents the possibility degree that a model of φ corresponds to the real word. Namely, this measure evaluates the consistency level of φ with respect to pieces of information.

A necessity measure N of a formula φ is defined as follows:

$$N(\varphi) = 1 - \Pi(\neg\varphi),$$

which corresponds to the certainty degree associated with φ from available pieces of information.

An important notion that can be derived from a possibility measure is the concept of possibility distribution. A possibility distribution π is a mapping from Ω to the interval $[0, 1]$. From a possibility measure Π, π is simply defined by $\forall \omega \in \Omega$, $\pi(\omega) = \Pi(\phi_\omega)$ where ϕ_ω is the formula which has only one model which is ω. And conversely, Π can be simply obtained from π as follows: $\Pi(\varphi) = \max\{\pi(\omega) : \omega \models \varphi\}$. The possibility degree $\pi(\omega)$ represents the compatibility of ω with available pieces of information. By convention, $\pi(\omega) = 1$ means that ω is totally possible, and $\pi(\omega) = 0$ means that ω is impossible. When $\pi(\omega) > \pi(\omega')$, ω is a preferred to ω' for being the real state of the world. A possibility distribution π is said to be *normalized* if there exists at least one interpretation which is consistent with available pieces of information, namely:

$$\exists \omega \in \Omega, \quad \pi(\omega) = 1.$$

2.2.1. Possibilistic conditioning

Conditioning [8] is a crucial notion for updating pieces of information (encoded by π) when a new evidence (completely sure information) e, is observed.

Several definitions have been proposed for possibilistic conditioning [9–11,8,7]. In next sections, we will only consider the so-called the min-based conditioning [8,7], defined by

$$\pi(\omega|\phi) = \begin{cases} 1 & \text{if } \pi(\omega) = \Pi(\phi) \quad \text{and} \quad \omega \models \phi, \\ \pi(\omega) & \text{if } \pi(\omega) < \Pi(\phi) \quad \text{and} \quad \omega \models \phi, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The existence of multiple definitions of conditioning, leads to several definitions of independence (see for instance [12–14]).

In this paper, we will only consider the non-interactivity relation [15,11]:

Let X, Y and Z be distinct subsets of V. X and Y are independent in the context of Z, iff $\forall x \in D_X$, $\forall y \in D_Y$, $\forall z \in D_Z$,

$$\pi(xy|z) = \min(\pi(x|z), \pi(y|z)).$$

2.3. Possibilistic logic

A possibilistic knowledge base [16] is a finite set of weighted formulas:

$$\Sigma = \{(\varphi_i, \alpha_i) : i = 1, \ldots, m\},$$

where α_i is the lower-boundary of the necessity degree $N(\varphi)$. Namely, $N(\varphi_i) \geqslant \alpha_i$. Formulas with a necessity degree equal to 0 are not explicitly represented in the knowledge base. The higher is the degree associated with a formula the more certain it is.

A possibilistic knowledge base Σ is said to be consistent if its classical support, obtained by forgetting the weights, is classically consistent.

Definition 1. Let Σ be a possibilistic knowledge base. The inconsistency degree of Σ, denoted $\mathrm{Inc}(\Sigma)$, is defined by

$$\mathrm{Inc}(\Sigma) = \max\{\alpha_i : \Sigma_{\geqslant \alpha_i} \models \perp\}, \tag{2}$$

where $\Sigma_{\geqslant \alpha_i}$ is a set of possibilistic formulas in Σ having a weight greater or equal to α_i.

$\mathrm{Inc}(\Sigma) = 0$ means that Σ is consistent.

Lang [17] proposed an algorithm to compute the inconsistency degree of Σ with a complexity equal to $\log_2 n\, SAT$ where n is the number of different valuations involved in Σ, and SAT is the propositional satisfiability test.

Possibilistic knowledge bases are compact representations of possibility distributions. Namely, we associate to each possibilistic knowledge base a unique possibility distribution [16] which is defined by

$\forall \omega \in \Omega$,

$$\pi_\Sigma(\omega) = \begin{cases} 1 & \text{if } \forall(\varphi_i, \alpha_i) \in \Sigma,\ \omega \models \varphi_i, \\ 1 - \max\{\alpha_i : \omega \not\models \varphi_i : (\varphi_i, \alpha_i) \in \Sigma\} & \text{otherwise.} \end{cases} \tag{3}$$

Example 1. Let $\Sigma = \{(a \vee b, \frac{1}{2}), (a, \frac{1}{4})\}$ be a possibilistic knowledge base.

Then, $\pi_\Sigma(ab) = \pi_\Sigma(a\neg b) = 1$; $\pi_\Sigma(\neg ab) = \frac{3}{4}$; $\pi_\Sigma(\neg a \neg b) = \frac{1}{2}$.

ab and $a\neg b$ are models of Σ. $\neg ab$ is preferred to $\neg a \neg b$ since it falsifies a lower-weighted formula $(a, \frac{1}{4})$ regarding $\neg a \neg b$ which falsifies $(a \vee b, \frac{1}{2})$.

Note that the possibility distribution π_Σ (associated with Σ) is not necessarily normalized. Namely, π_Σ is normalized if and only if Σ is consistent. Moreover, it can be checked that [16]:

$$\mathrm{Inc}(\Sigma) = 1 - \max_{\omega} \pi_\Sigma(\omega).$$

Definition 2. Two knowledge bases Σ and Σ' are said to be equivalent if and only if they have the same associated possibility distributions. Namely:

$$\forall \omega \in \Omega, \quad \pi_\Sigma(\omega) = \pi_{\Sigma'}(\omega).$$

Possibilistic knowledge bases can be simplified by applying subsumption which allows to delete (or to add) some formulas without lose of information.

The subsumption is defined as follows:

Definition 3. Let (φ, α) be a formula in Σ. Then (φ, α) is said to be subsumed by Σ if Σ and $\Sigma \backslash \{\varphi, \alpha\}$ are equivalent knowledge bases.

Namely, subsumed formulas are redundant formulas that can be removed or added without changing possibility distributions.

2.4. Standard possibilistic networks and algorithms

2.4.1. Standard possibilistic networks

We now define the standard possibilistic networks, denoted ΠG, which can be viewed as the counterparts of probabilistic bayesian networks [2,4,1]. A standard possibilistic network consists of:

- A graphical component: It is a directed acyclic graph (DAG). Nodes represent variables and edges correspond to "causality" links.
- A quantitative component: Uncertainty is represented at each node by local conditional possibility distributions.

The normalization conditions on each variable A of the graph are:

- If A is a root, namely $U_A = \emptyset$, then we provide $\pi(a)$ and $\pi(\neg a)$ with $\max(\pi(a), \pi(\neg a)) = 1$.
- If A has parents, namely $U_A \neq \emptyset$, then we provide $\pi(a|u_A)$ and $\pi(\neg a|u_A)$, with $\max(\pi(a|u_A), \pi(\neg a|u_A)) = 1$, for each $u_A \in D_{U_A}$, where D_{U_A} is the cartesian product of domains of variables which are parents of A.

Possibilistic networks are also compact representations of possibility distributions. In fact, we can follow the same procedure as in probability theory by expressing a joint possibility distribution as a combination of conditional possibility distributions (chain rule). Namely,

$$\pi_{\Pi G}(a_1, \ldots, a_n) = \min_{i=1,\ldots,n} \pi(a_i|u_{A_i}), \tag{4}$$

where a_i is an instance of A_i and $u_{A_i} \subseteq \{a_1, \ldots, a_n\}$ is an element of the cartesian product of domains associated with variables U_{A_i} which are parents of A_i.

Example 2. Fig. 1 gives an example of a possibilistic network. Table 1 provides local conditional possibility distributions of each node given its parents.

Using the possibilistic chain rule, the possibility degree of $\pi_{\mathrm{HG}}(\neg ab\neg cd)$ is computed as follows:

$$\pi_{\mathrm{HG}}(\neg ab\neg cd) = \min(\pi(\neg a), \pi(b|\neg a), \pi(\neg c|\neg a), \pi(d|b\neg c)) = \min\left(1, \frac{1}{4}, 1, 1\right) = \frac{1}{4}.$$

2.4.2. Inference algorithm in multiply connected possibilistic networks

Propagation algorithms aim to establish a posteriori possibility distribution of each node A given some evidence on a set of variables E. When DAGs are singly connected then the propagation algorithm is polynomial. In this section, we only focus on multiply connected graphs.

A well-known propagation algorithm for multiply connected graphs proceeds to a transformation of the initial graph into a junction tree. The main steps of the junction tree construction are:

– Moralisation of the initial DAG: Create an undirected graph from the initial graph and add links between parents of a common variable.

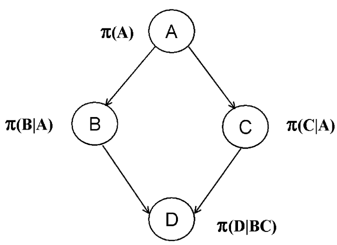

Fig. 1. Example of possibilistic causal network ΠG.

Table 1
Local conditional possibility distributions associated with DAG of Fig. 1

a	$\frac{1}{4}$
$\neg a$	1

$B\|A$	a	$\neg a$
b	$\frac{1}{4}$	$\frac{1}{4}$
$\neg b$	1	1

$C\|A$	a	$\neg a$
c	1	$\frac{1}{2}$
$\neg c$	$\frac{3}{4}$	1

$D\|BC$	bc	$\neg bc$	else
d	1	$\frac{1}{4}$	1
$\neg d$	$\frac{1}{2}$	1	1

– Triangulation of the moral graph: Consists of adding edges to connect non-adjacent nodes in cycles of length four or greater.
– Building a junction tree from the triangulated moral graph: Consists of the junction tree construction by choosing the appropriate clusters and separators from the triangulated graph.

The main idea of these steps is to transform multiply connected graphs into tree structures by gathering some variables in a same node. Each node of the resulting tree, called cluster, is a set of variables. Common variables of two adjacent clusters are grouped into another type of node, called a separator.

Fig. 2 gives an example of a junction tree associated with the DAG of Fig. 1 (there are two clusters $\{ABC\}$ and $\{BCD\}$ and one separator $\{BC\}$ which is the intersection of the two clusters).

The propagation algorithm is then applied on this resulting structure. The idea is to require that adjacent clusters sharing common variables should have the same marginal distributions with respect to these common variables namely on their separator. The main steps of the junction tree propagation algorithm are (for more details see [3,18]):

- *Step S1: Standard initialization*
  This step initializes possibility distributions associated with clusters and separators using local possibility distributions in the initial DAG.
  – For each cluster $C_i : \pi^I_{C_i} \leftarrow \mathbb{1}$, where $\mathbb{1}$ is a possibility distribution where all elements have a highest possibility degree 1.
  – For each separator $S_{ij} : \pi^I_{S_{ij}} \leftarrow \mathbb{1}$.
  – For each variable A, select a cluster C_i containing $A \cup U_A$ and update its possibility distributions as follows: $\pi^I_{C_i} : \pi^I_{C_i} \leftarrow min(\pi^I_{C_i}, \Pi(A|U_A))$.
- *Step S2: Standard handling of evidence*
  If one has an evidence $E = \{e_1, \ldots, e_n\}$, then for each e_i select a cluster C_i containing E_i, and update its possibility distribution as follows:

$$\pi^I_{C_i}(\omega) = \min(\pi^I(\omega), \pi_{e_i}(\omega)),$$

  where $\pi_{e_i}(\omega)$ is defined:

$$\pi_{e_i}(\omega) = \begin{cases} 1 & \text{if } \omega \models e_i, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

After initialization and handling evidence, steps S3 and S4 consist of updating local distributions associated with C_j and S_{ij} (separator) for each message sent from C_i to C_j (adjacent clusters). The sequence of updating is not important. Steps S3 and S4 are repeated until the junction tree is globally consistent, namely adjacent clusters should have same marginal distributions with respect to common variables, namely:

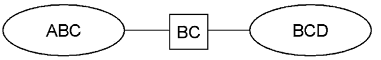

Fig. 2. Junction tree associated with graph ΠG of Fig. 1.

$$\max_{C_i \setminus S_{ij}} \pi^t_{C_i} = \pi^{t+1}_{S_{ij}} = \max_{C_j \setminus S_{ij}} \pi^t_{C_j}.$$

- *Step S3: Standard updating of separators*
  The distribution associated with a separator S_{ij}, denoted $\pi_{S_{ij}}$, is updated using the local possibility distribution π_{C_i} associated with the cluster C_i sending the message:
  $$\pi^{t+1}_{S_{ij}} \leftarrow \max_{C_i \setminus S_{ij}} \pi^t_{C_i}, \tag{6}$$
  where t (respectively $t+1$) means the moment before (respectively after) sending the message from C_i to C_j through S_{ij}.
- *Step S4: Standard updating of clusters*
  Each cluster updates its possibility distribution, denoted $\pi^{t+1}_{C_j}$, when receiving a message from its adjacent separator as follows:
  $$\pi^{t+1}_{C_j} \leftarrow min\left(\pi^t_{C_j}, \pi^t_{S_{ij}}\right). \tag{7}$$
- *Step S5: Computing queries*
  When the junction tree is consistent, computing $\Pi(A = a|E)$ consists in selecting any cluster containing A and marginalizing Π_{C_i} on A:
  $$\Pi(A = a|E) = \Pi_{C_i}(A = a).$$

3. Hybrid representation of possibilistic networks

Pieces of information can be provided either in terms of possibilistic knowledge bases or in terms of conditional possibility distributions (if the size of universe of discourse is reasonable). They can also be represented either using graphical structures or logic-based structures.

In this section, we describe our hybrid representation of uncertain information in possibilistic networks.

The aim of this representation is to take advantage of these two possible representation formats. Graphical representation is used to take advantage of independence relations, and logic-based representation is used to have compact representation of possibility distributions.

Local uncertainty is no longer represented by conditional possibility distributions but by possibilistic knowledge bases. As it is said in Section 1, the main advantage of this representation concerns space complexity. Indeed, during the junction tree construction, it may happen that the size of clusters can be very large. In this case, it can be impossible to explicitly construct possibility distributions associated with clusters.

Hybrid possibilistic causal networks, denoted HG, are characterized by:

- *A graphical component* which is represented by a DAG (like standard possibilistic causal networks), denoted HG, that allows to represent independence relationships.
- *A quantitative component* which encodes uncertainties. It associates to each node a local knowledge base instead of a conditional possibility distribution. Namely, at each node A_i, one provides a possibilistic knowledge base Σ_{A_i} which represents a local knowledge base regarding the variable A and its parents.

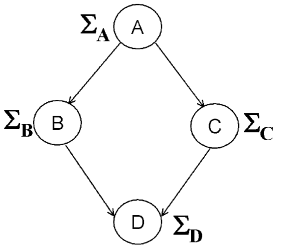

Fig. 3. Example of hybrid graph HG.

Table 2
Joint possibility distribution Π_{HG} associated with HG of Fig. 3

ω	π_{HG}	ω	π_{HG}	ω	π_{HG}	ω	π_{HG}
$abcd$	$\frac{1}{4}$	$a\neg bcd$	$\frac{1}{4}$	$\neg abcd$	$\frac{1}{4}$	$\neg a\neg bcd$	$\frac{1}{2}$
$abc\neg d$	$\frac{1}{4}$	$a\neg bc\neg d$	$\frac{1}{4}$	$\neg abc\neg d$	$\frac{1}{4}$	$\neg a\neg bc\neg d$	$\frac{1}{2}$
$ab\neg cd$	$\frac{1}{4}$	$a\neg b\neg cd$	$\frac{1}{4}$	$\neg ab\neg cd$	$\frac{1}{4}$	$\neg a\neg b\neg cd$	1
$ab\neg c\neg d$	$\frac{1}{4}$	$a\neg b\neg c\neg d$	$\frac{1}{4}$	$\neg ab\neg c\neg d$	$\frac{1}{4}$	$\neg a\neg b\neg c\neg d$	1

From each local knowledge base Σ_{A_i} we can compute the local possibility distribution $\pi_{\Sigma_{A_i}}(A_i \wedge U_i)$ using Eq. (3). Hybrid graphs are also compact representations of joint possibility distributions. Namely, a possibility distribution associated with a hybrid possibilistic network HG is defined by

$$\forall\omega,\ \pi_{\mathrm{HG}}(\omega) = \min_{A_i \in V} \pi_{\Sigma_{A_i}}(\omega), \tag{8}$$

where $\{\pi_{\Sigma_{A_i}} : A_i \in V\}$ are possibility distributions associated with $\{\Sigma_{A_i} : A_i \in V\}$ using Eq. (3).

Example 3. Fig. 3 provides an example of hybrid possibilistic graphs.

Let Σ_A, Σ_B and Σ_C be the local possibility knowledge bases respectively on A, B and C:

$$\Sigma_A = \left\{\left(\neg a, \frac{3}{4}\right)\right\},$$

$$\Sigma_B = \left\{\left(\neg a \vee \neg b, \frac{3}{4}\right), \left(a \vee \neg b, \frac{3}{4}\right)\right\},$$

$$\Sigma_C = \left\{\left(a \vee \neg c, \frac{1}{2}\right), \left(\neg a \vee c, \frac{1}{4}\right)\right\},$$

$$\Sigma_D = \left\{\left(b \vee \neg c \vee \neg d, \frac{3}{4}\right), \left(\neg b \vee \neg c \vee d, \frac{1}{2}\right)\right\}.$$

The joint possibility distribution associated with HG is given in Table 2.

4. Propagation algorithms on hybrid junction trees

This section presents the counterpart of the junction tree propagation algorithm for a hybrid representation.

Standard representation and hybrid representation have the same graphical structure, namely DAGs. Therefore, the transformation from a DAG to a junction tree in hybrid representation exactly follows the same steps of Section 2 (moralisation, triangulation, etc.). One of the limits of junction tree algorithm is that the transformation step of initial multiply connected graph can produce clusters with a great number of variables. In that case, it may be impossible to get local joint possibility distributions on clusters.

Before introducing the propagation algorithm in the hybrid framework, we need to present the notion of prioritized forgetting (see [5]) which allow to give the syntactic counterpart of the marginalization process.

4.1. Prioritized forgetting: a syntactic computation of marginalization

Lin and Reiter [19] proposed an approach allowing variable domain restriction in propositional knowledge bases (see [20,21] for details). Variable forgetting (also known as projection or marginalization) is defined as:

Definition 4. Let K be a propositional knowledge base and X be a propositional variable set. The forgetting of X in K, denoted *forget*(K, X), is equivalent to a propositional formula that can be inductively defined as follows:

- $\text{forget}(K, \emptyset) = K$,
- $\text{forget}(K, \{x\}) = K_{x\leftarrow\bot} \vee K_{x\leftarrow\top}$,
- $\text{forget}(K, X \cup \{x\}) = \text{forget}(\text{forget}(K, X), \{x\})$,

where $K_{x\leftarrow\bot}$ (respectively $K_{x\leftarrow\top}$) refers to K in which we affect false (respectively true) value to each occurrence of x. By $K_i \vee K_j$ we mean the set $\{(\varphi_i \vee \psi_j) : \varphi_i \in K_i \text{ and } \psi_j \in K_j\}$.

This approach is defined for classical propositional logic. We present an extension of this definition, called prioritized forgetting, which deals with possibilistic knowledge bases. First, we need to precise the notion of disjunction.

Definition 5. Let Σ_1 and Σ_2 be two possibilistic knowledge bases. The disjunction of two bases in possibilistic framework, denoted by $\bigcirc\!\!\!\!\vee$, is defined as follows:

$$\Sigma_1 \bigcirc\!\!\!\!\vee \Sigma_2 = \{(\varphi_i \vee \psi_j, \min(\alpha_i, \beta_j)) : (\varphi_i, \alpha_i) \in \Sigma_1 \text{ and } (\psi_j, \beta_j) \in \Sigma_2\}.$$

Prioritized forgetting, denoted pforget, can then be defined as a simple extension of Definition 4 by replacing classical disjunction ($\vee$) by the operator $\bigcirc\!\!\!\!\vee$ defined above:

Definition 6. Let Σ be a possibilistic knowledge base and X be a variable set. The prioritized forgetting of X in Σ, denoted $\text{pforget}(\Sigma, X)$, is equivalent to a possibilistic knowledge base defined by

- $\text{pforget}(\Sigma, \emptyset) = \Sigma$,
- $\text{pforget}(\Sigma, \{x\}) = K_{x\leftarrow\bot} \bigcirc\!\!\!\!\vee K_{x\leftarrow\top}$,
- $\text{pforget}(\Sigma, X \cup \{x\}) = \text{pforget}(\text{pforget}(\Sigma, X), \{x\})$.

Prioritized forgetting allows to syntacticly capture the base associated with marginal distributions. More precisely:

Proposition 1. *Let Σ be a possibilistic knowledge base and π its associated distribution. Let X be a set of variables. Then the possibility distribution associated with pforget*(Σ, X), *denoted* $\pi_{pforget(\Sigma, X)}$ *is equivalent to the possibility distribution resulting from the semantic marginalization of X in* Σ:

$$\pi_{\mathrm{pforget}(\Sigma,X)} = \max_{V \setminus X} \pi_{\Sigma}. \tag{9}$$

Proof (*Sketch*). For sake of simplicity (without loss of generality) we assume that Σ is under clausal form and is free of tautologies. We also assume that clauses do not contain repeated literals.

To show that $\pi_{\mathrm{pforget}(\Sigma,X)}(\cdot) = \max_{V \setminus X}\pi_{\Sigma}(\cdot)$ it is enough to show that the equality holds for a singleton.

Let A be a variable. We need to show that:

$$\forall \omega \in \times_{B \in V \setminus \{A\}} D_B, \ \ \pi_{\mathrm{pforget}(\Sigma,\{A\})}(\omega) = \max(\pi_{\Sigma}(\omega a), \pi_{\Sigma}(\omega \neg a)). \tag{10}$$

In [22] it has been shown that if Σ_1 and Σ_2 are two possibilistic knowledge bases then $\forall \omega, \pi_{\Sigma_1 \bigcirc\!\!\!\!\vee \Sigma_2}(\omega) = \max(\pi_{\Sigma_1}(\omega), \pi_{\Sigma_2}(\omega))$. This means that $\pi_{\mathrm{pforget}(\Sigma,\{A\})}(\omega) = \max(\pi_{\Sigma_{A \leftarrow \top}}(\omega), \pi_{\Sigma_{A \leftarrow \bot}}(\omega))$.

Therefore to show (10) it is enough to show that $\forall \omega \in \times_{B \in V \setminus \{A\}} D_B$, $\pi_{\Sigma}(\omega a) = \pi_{\Sigma_{A \leftarrow \top}}(\omega)$ and $\pi_{\Sigma}(\omega \neg a) = \pi_{\Sigma_{A \leftarrow \bot}}(\omega)$.

The proof of $\pi_{\Sigma}(\omega a) = \pi_{\Sigma_{A \leftarrow \top}}(\omega)$ (respectively $\pi_{\Sigma}(\omega \neg a) = \pi_{\Sigma_{A \leftarrow \bot}}(\omega)$ is immediate. Let $(\phi, \alpha) \in \Sigma$ be a possibilistic formula. If ϕ is falsified by ωa then $\phi_{A \leftarrow \top}$ is falsified by ω and conversely.

Indeed, if ϕ is a clause that does not contain a then ϕ is equivalent to $\phi_{A \leftarrow \top}$, hence ωa falsifies ϕ iff ω falsifies $\phi_{A \leftarrow \top}$.

Now, if ϕ is of the form $\psi \vee a$ then $\phi_{A \leftarrow \top}$ is equivalent to a tautology, hence ωa and ω satisfies respectively ϕ and $\phi_{A \leftarrow \top}$.

Lastly, if ϕ is of the form $\psi \vee \neg a$ then $\phi_{A \leftarrow \top}$ is equivalent to ψ, hence ωa falsifies $\psi \vee \neg a$ iff ω falsifies ψ. □

4.2. Propagation algorithm

The main steps of the new hybrid junction tree (HJT) algorithm are summarized in the following figure:

Procedure Hybrid junction tree propagation
Begin
– Junction tree construction from the initial graph.
– Apply step H1: Hybrid initialization,
– Apply step H2: Hybrid handling of evidence (if $e \neq \emptyset$),
While (Junction tree is not consistent) **do**

– Apply step H3: Hybrid updating separators,
– Apply step H4: Hybrid updating clusters.
done
– Apply step H5: Hybrid computing queries.
Steps H1, H2, H3, H4 and H5 are detailed below.
End

Steps (H1–H5) are the counterparts of steps (S1–S5) using hybrid representation based on possibilistic knowledge bases.

4.3. Step H1: Hybrid initialization

This step consists of initializing the junction tree by assigning knowledge bases to clusters and separators.

- An empty knowledge base Σ_{C_i} is first assigned to each cluster C_i:
  $\Sigma_{C_i} \leftarrow \emptyset$.
- An empty knowledge base $\Sigma_{S_{ij}}$ is also assigned to each separator S_{ij}:
  $\Sigma_{S_{ij}} \leftarrow \emptyset$.
- For each variable A, select a cluster C_i containing $\{A\} \cup U_A$ and add to the knowledge base Σ_{C_i} the possibilistic base Σ_A associated with A:

$$\Sigma_{C_i} \leftarrow \Sigma_{C_i} \cup \Sigma_A.$$

Proposition 2. *The joint possibility distribution π_{HJT} associated with the junction tree HJT after the initialization step is equivalent to the initial joint distribution π_{HG} associated with the initial graph HG (before the graphical transformation):*

$$\pi = \pi_{\mathrm{HG}} = \pi^I_{\mathrm{HJT}} = \min_{i=1,\ldots,n} \pi^I_{\Sigma_{C_i}},$$

where π^I_{HJT} is the possibility distribution associated with Σ_{HJT} after the initialization step, and π_{HG} is the distribution associated with the initial graph.

This result can be immediately proved since $\Sigma^I_{\mathrm{HJT}} = \bigcup_{i=1}^n \Sigma^I_{C_i} = \bigcup_{A_j \in \mathrm{HG}} \Sigma_{A_j} = \Sigma_{\mathrm{HG}}$, where Σ_{A_j} is the possibilistic knowledge base associated with the variable A_j ($A_j \in$ HJT).

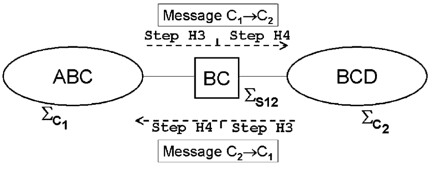

Fig. 4. Message passing in the junction tree HJT.

Example 4. Given the junction tree of Fig. 4 obtained from the graph HG in Fig. 3 and the local knowledge bases of the initial graph HG presented in Example 3, local knowledge bases associated with clusters after the initialization step are:

$$\Sigma_{C_1} = \Sigma_A \cup \Sigma_B \cup \Sigma_C = \left\{ \left(\neg a, \frac{3}{4}\right), \left(a \vee \neg b, \frac{3}{4}\right), \left(a \vee \neg c, \frac{1}{2}\right) \right\}$$
$$\times \left(\left(\neg a \vee c, \frac{1}{4}\right) \text{and} \left(\neg a \vee \neg b, \frac{3}{4}\right) \text{ are subsumed formulas} \right).$$
$$\Sigma_{C_2} = \Sigma_D = \left\{ \left(b \vee \neg c \vee \neg d, \frac{3}{4}\right), \left(\neg b \vee \neg c \vee d, \frac{1}{2}\right) \right\}.$$

Let us consider the interpretation $\omega = \neg ab\neg cd$. We have:

$\pi_{\mathrm{HJT}}(\neg ab\neg cd) = \min(\pi_{\Sigma_{C_1}}(\neg ab\neg c), \pi_{\Sigma_{C_2}}(b\neg cd)) = \frac{1}{4}$ which is the same as $\pi_{\mathrm{HG}}(\neg ab\neg cd)$ computed in Table 2.

4.4. Step H2: Hybrid handling evidence (if $e \neq \emptyset$)

To handle the evidence, we should update the initial local knowledge base at the level of the initial hybrid possibilistic graph HG.

Indeed, we should encode the evidence e for each observed variable $E = e$, by adding the possibilistic formula $(e, 1)$ to a local knowledge base Σ_{C_i} where C_i is a cluster containing E.

More precisely, if one has an evidence $E_i = e_i$, then select a cluster C_i containing E_i, and update its possibility distribution as follows:

$$\Sigma_{C_i} = \Sigma_{C_i} \cup \Sigma_{e_i},$$

where Σ_{e_i} is defined:

$$\Sigma_{e_i} = \begin{cases} (e_i, 1) & \text{if } E_i \text{ is observed to } e_i, \\ \emptyset & \text{otherwise.} \end{cases} \tag{11}$$

After the hybrid handling evidence step (H2), the distribution $\pi_{\Sigma_{C_i}}$ associated with Σ_{C_i} (C_i: updated cluster) is the same as the distribution π_{C_i} computed using the step S2 during the standard algorithm process. Namely:

$$\pi^{\mathrm{H2}}_{\Sigma_{C_i}} = \pi^{\mathrm{S2}}_{C_i}.$$

In fact,

$$\pi^{\mathrm{H2}}_{\Sigma_{C_i}} = \min(\pi^{\mathrm{H1}}_{\Sigma_{C_i}}, \pi_{\Sigma_{e_i}})$$

(union of the knowledge bases is the syntactic counterpart of the min operator on their possibility distributions)

$$= \min(\pi^{\mathrm{S1}}_{C_i}, \pi_{\Sigma_{e_i}})$$

Since $\pi_{\Sigma_{e_i}}$ is defined by (using (3)):

$$\pi_{\Sigma_{e_i}}(\omega) = \begin{cases} 1 & \text{if } \omega \models e_i, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Then, $\pi_a = \pi_{\Sigma_a}$ and $\pi^{\mathrm{H2}}_{\Sigma_{C_i}} = \pi^{\mathrm{S2}}_{C_i}$.

4.5. Step H3: Hybrid updating separators

After the initialization and handling evidence steps, messages are sent between clusters in order to guarantee the consistency conditions. If, for instance, for given two clusters C_i and C_j, we have:

$$\max_{C_i \setminus S_{ij}} \pi_{C_i} \neq \max_{C_j \setminus S_{ij}} \pi_{C_j},$$

then C_i and C_j should update their knowledge bases iteratively. The following two elementary steps are repeated until reaching consistency:

- A separator S_{ij} computes its knowledge base from C_i (respectively C_j).
- A cluster C_j (respectively C_i) updates its knowledge base taking into account the knowledge base of the separator previously computed.

The knowledge base $\Sigma_{S_{ij}}$, associated with a separator S_{ij}, represents the restriction (marginalization) of the base Σ_{C_i} (respectively Σ_{C_j}) on common variables in the separator S_{ij}. This knowledge base is immediately obtained thanks to Proposition 1.

Corollary 1. *After receiving a message from C_i, the possibilistic knowledge base $\Sigma_{S_{ij}}$ associated with S_{ij} is updated as follows*:

$$\Sigma_{S_{ij}} = \text{pforget}(\Sigma_{C_i}, C_i \setminus S_{ij}).$$

Example 5. We assume that $E = \emptyset$. Given the hybrid junction tree HJT in Fig. 4 and the local knowledge bases Σ_{C_1} and Σ_{C_2} in Example 4. Let us compute the knowledge base $\Sigma_{S_{12}}$, associated with the separator S_{12} using Σ_{C_1}. This leads to forgetting the variable A. Let us apply the definition of pforget:

- $\Sigma_{a \leftarrow \bot} = \left\{ \left(\neg b, \frac{3}{4}\right), \left(\neg c, \frac{1}{2}\right) \right\}$
- $\Sigma_{a \leftarrow \top} = \left\{ \left(\bot, \frac{3}{4}\right) \right\}$

$$\Sigma_{S_{12}} = \text{pforget}(\Sigma_{C_1}, \{A\}) = \left\{ \left(\neg b, \frac{3}{4}\right), \left(\neg c, \frac{1}{2}\right) \right\}.$$

4.6. Step H4: Hybrid updating clusters

Proposition 3. *When receiving a message from C_i to C_j (via S_{ij}), the possibilistic knowledge base Σ_{C_i} associated with C_i is updated as follows*:

$$\Sigma_{C_i} = \Sigma_{C_i} \cup \Sigma_{S_{ij}}, \tag{13}$$

where the union is in fact the syntactical equivalent of possibilistic distributions combination by the minimum operator.

The global distribution π_{HJT} associated with the junction tree HJT is the same as the initial distribution since $\pi = \pi_{\mathrm{HG}} = \pi^I_{\mathrm{HJT}} = \min_{i=1,\ldots,n} \pi_{\Sigma_{C_i}}$. The proof can be easily checked since $\Sigma_{C_i} \cup \Sigma_{C_j} \cup \Sigma_{S_{ij}}$ is equivalent to $\Sigma_{C_i} \cup \Sigma_{C_j}$.

The steps of updating separators and clusters knowledge bases are repeated until reaching stability (global consistency) in the junction tree. Formally, HJT is consistent if $\forall i,j$, we have:

$$\Sigma_{S_{ij}} = \text{pforget}(\Sigma_{C_i}, C_i \setminus S_{ij}) = \text{pforget}(\Sigma_{C_j}, C_j \setminus S_{ij}). \tag{14}$$

Proposition 4. *The joint possibility distribution π_{HJT} associated with the junction tree after sending all messages (updating clusters and updating separators' steps) is equivalent to the initial distribution conditioned with evidence E:*

$$\pi(.|E) = \pi_{\text{HG}}(.|E) = \pi_{\text{HJT}}(.|E).$$

Example 6. We assume that $E = \emptyset$. The knowledge base Σ_{C_2} associated with the cluster C_2 after receiving $\Sigma_{S_{12}}$ is:

$\Sigma_{C_2} = \Sigma_{C_2} \cup \Sigma_{S_{12}} = \{(b \vee \neg c \vee \neg d, \frac{3}{4}), (\neg b \vee \neg c \vee d, \frac{1}{2}), (\neg b, \frac{3}{4}), (\neg c, \frac{1}{2})\}$ which is equivalent to $\Sigma_{C_2} = \{(b \vee \neg c \vee \neg d, \frac{3}{4}), (\neg b, \frac{3}{4}), (\neg c, \frac{1}{2})\}$.

At the end of propagation process, we obtain the following local knowledge bases:

- $\Sigma_{C_1} = \{(\neg a, \frac{3}{4}), (\neg b, \frac{3}{4}), (\neg c, \frac{1}{2})\}$.
- $\Sigma_{C_2} = \{(\neg b, \frac{3}{4}), (\neg c, \frac{1}{2}), (b \vee \neg c \vee \neg d, \frac{3}{4})\}$.

It can be checked that HJT is consistent.

Moreover, the possibility distribution associated with HJT is equivalent to the initial possibility distribution. For instance,

$$\pi_{\text{HJT}}(\neg ab\neg cd) = \min(\pi_{\Sigma_1}(\neg ab\neg cd), \pi_{\Sigma_1}(\neg ab\neg cd)) = \min\left(\frac{1}{4}, \frac{1}{4}\right) = \frac{1}{4}.$$

4.7. H5: Hybrid computing queries

The propagation algorithm aims to evaluate the conditional possibility distribution associated with a variable A after or not observing some other variables. Thus, computing $\Pi(A|E)$ is done syntactically using possibilistic inference once the junction tree is consistent:

Proposition 5. *Let Σ be a possibilistic knowledge base. Let a be an instance of A. Then,*

$$\pi(a) = 1 - Inc(\Sigma \cup \{(a, 1)\}),$$

where $Inc(\Sigma \cup \{(a,1)\})$ is the inconsistency degree of $\Sigma \cup \{(a, 1)\}$. An approach for computing the inconsistency degree Inc was proposed in [16].

Hence, once the junction tree is consistent, computing $\pi(a|E)$ comes down to select any cluster C_i that contains A and compute $1 - \text{Inc}(\Sigma_{C_i} \cup \{(a, 1)\})$ which gives the possibility degree of a given E.

Example 7. We recall that $E = \emptyset$. Assume that we are interested to compute $\pi(a)$. C_1 is a cluster which contains the variable A. Then, $\pi(a) = 1 - \text{Inc}(\Sigma_{C_1} \cup \{(a, 1)\}) = 1 - \frac{3}{4} = \frac{1}{4}$.

5. Hybrid possibilistic networks vs standard possibilistic frameworks

This section shows that hybrid possibilistic networks encode both possibilistic networks and possibilistic logic.

We start by considering standard possibilistic causal networks ΠG where uncertainty is represented at the level of nodes by conditional possibility distributions. The construction of local possibilistic knowledge bases is similar to the one used in [23].

First, hybrid network will have the same structure as the standard possibilistic network. Let A be a binary variable and a_i be an instance of this variable. Let $\pi(a_i|u_i)$ be a local possibility degree associated with a_i given u_i, where u_i is an element of cartesian product of its parents (U_A) domains. Let us associate the following possibilistic knowledge base with the node A:

$$\Sigma_A = \{(\neg a_i \vee \neg u_i, \alpha_i) : \alpha_i = 1 - \pi(a_i|u_i) \neq 0\}. \tag{15}$$

It is easy to check that the conditional possibilities are recovered from Σ_A using Eq. (3). $\pi_{\Sigma_A}(a_i \wedge u_i) = 1 - \alpha_i$ holds since interpretations that satisfy $a_i \wedge u_i$ falsify $(\neg a_i \vee \neg u_i, \alpha_i)$. Normalization is also satisfied since $\pi_{\Sigma_A}(u_i) = \max(\pi_{\Sigma_A}(a_i \wedge u_i), \pi_{\Sigma_A}(\neg a_i \wedge u_i)) = 1$ (i.e. $\neg a_i \wedge u_i$ satisfies $(\neg a_i \vee \neg u_i, \alpha_i)$).

Then, it can be easily proved that $\forall \omega$,

$$\pi_{\Pi G}(\omega) = \pi_{\mathrm{HG}}(\omega), \tag{16}$$

where $\pi_{\Pi G}$ and π_{HG} are obtained by using Eqs. (4) and (8).

Example 8. Let us build a hybrid possibilistic causal network HG from standard possibilistic causal network ΠG of Example 2 by associating knowledge bases to each node using (15). Uncertainty at the level of nodes A, B and C (binary variables) is represented by possibilistic knowledge bases Σ_A, Σ_B and Σ_C as follows:

$$\Sigma_A = \left\{\left(a, \frac{1}{4}\right)\right\},$$

$$\Sigma_B = \left\{\left(\neg b, \frac{1}{2}\right)\right\},$$

$$\Sigma_C = \left\{\left(\neg a \vee \neg b \vee c, \frac{3}{4}\right), \left(a \vee \neg b \vee \neg c, \frac{1}{2}\right), \left(\neg a \vee b \vee c, \frac{1}{4}\right), \left(a \vee b \vee c, \frac{1}{4}\right)\right\}.$$

We can check that the joint possibility distribution $\pi_{\mathrm{HG}}(ABC)$ using Eq. (8) is the same as the one given in Example 2. For instance, $\pi_{\mathrm{HG}}(\neg ab\neg c) = \min(\Sigma_A(\neg ab\neg c), \Sigma_B(\neg ab\neg c), \Sigma_C(\neg ab\neg c)) = \min\left(\frac{3}{4}, \frac{1}{2}, 1\right) = \frac{1}{2} = \pi_{\Pi G}(\neg ab\neg c)$.

The encoding of possibilistic knowledge base Σ is immediate. Its associated hybrid possibilistic network HG can be constructed in the following way:

- Select an arbitrary variable A. Assign to A the knowledge base Σ.
- For each variable $B \neq A$, add a link from B to A.
- Assign an empty possibilistic knowledge base to B.

Then,

$$\pi_{\mathrm{HG}}(\omega) = \pi_{\Sigma}(\omega), \tag{17}$$

since $\pi_{\mathrm{HG}} = \pi_{\Sigma_A}$ and $\Sigma_A = \Sigma$.

6. Experimental results

The hybrid junction tree algorithm proposed in Section 4 can be used as an alternative propagation algorithm to the standard one. The idea of the implemented algorithm is the following: if in the initialization step, the size of cluster is not preventing to use possibility distributions (great variable number in clusters) then standard propagation steps (S1–S5) described in Section 2 are used. Now, if it is impossible to represent distribution over clusters, then we use alternative steps (H1–H5).

The main steps of the implemented algorithm is summarized as the following:

It is clear that our algorithm is an improvement of standard junction tree propagation, since steps H1–H5 are run only if it is not possible to initialize the junction tree with explicit local conditional possibility distributions. In this section, we present experimental results for the proposed possibilistic propagation algorithm. These experimentations show that our algorithm is a real improvement, since we identify several examples where standard junction tree blocks, while our algorithm provides answers.

The experimentation was conducted on sets of randomly generated possibilistic networks. DAGs are generated randomly by varying number of nodes and the maximum number of parents. We define links ratio to be the average number of links per node in the graph. Local conditional distributions on each node in the context of its parents are also generated randomly respecting the normalization constraints. It is well-known that the performance of standard junction tree does not depend on numerical degrees assigned to interpretations. In hybrid networks, the performance of the propagation algorithms depends on possibility distributions. The smaller is the number of interpretations having possibility degrees different from 0 and 1, the more efficient is the algorithm. In our experimentation, the number of interpretations having possibility degree different of 0 and 1 is around 25%. The experimentations show that with networks containing 35 (respectively 40, 50, 60) nodes, it begins to be impossible to initialize local distributions at the level of clusters with links ratio around 4.45 (respectively 3.55, 2.72, 1.78).

Procedure Hybrid junction tree propagation
Begin
- Junction tree construction from the initial graph
- Apply step S1: Standard initialization

If (Standard initialization succeeds) **then**
- Apply step S2: Standard handling of evidence,

While (Junction tree is not consistent) **do**
- Apply step S3: Standard updating separators,
- Apply step S4: Standard updating clusters.

done
- Apply step S5: Standard computing queries.

else

Table 3
Experimental results

Nb nodes	Average ratio of links/nodes	JT algo errors (%)	Average time hybrid (s)	Hybrid algo errors (%)
30	4.32	0	0.91s	0
35	4.42	8	126.45s	0
40	4.58	55	240.97s	2
45	4.55	87	393.37s	2
50	4.67	≈100	1535.48	15

– For each variable A, compute local knowledge base π_{Σ_A},
– Apply step H1: Hybrid initialization,
While (Junction tree is not consistent) **do**
  – Apply step H2: Hybrid handling of evidence,
  – Apply step H3: Hybrid updating separators,
  – Apply step H4: Hybrid updating clusters.
done
– Apply step H5: Hybrid computing queries.
Steps H1, H2, H3, H4 and H5 are detailed below.
end If
End

Results in Table 3 are obtained by fixing the maximum number of parents to 10. In most cases, we observe that hybrid junction tree algorithm provides a response. Our new algorithm can only be limited by the running-time but never blocks. We chose to set a time-limit equal to 10000 s. Clearly, in many examples when standard possibilistic networks block our algorithm provides answers. Particularly for networks with 50 nodes, standard junction tree algorithm blocks for basically each generated network.

7. Conclusion

This paper provides a new representation of possibility networks, where conditional possibility distributions are compactly represented by local possibilistic knowledge bases. We have shown that standard possibilistic graphs can be equivalently encoded in hybrid possibilistic graphs.

We then extended the notion of forgetting variables introduced in [19–21], and showed that this extension indeed allows the computation of marginalized knowledge base.

An adaptation of junction tree algorithm is provided. When uncertainty on clusters is described by possibilistic knowledge bases, our algorithm improves standard junction tree propagation algorithms.

We also established the relationships between possibilistic frameworks and hybrid possibilistic networks. We have shown that hybrid possibilistic graphs can be encoded into possibilistic logic and into standard possibilistic networks.

Lastly, we provide experimental studies where examples, which are blocked by standard junction tree algorithm, are solved using our algorithm based on hybrid representation of possibilistic networks.

The idea of combining logical-based representation and graphical models have been previously considered by several authors [24–26,7]. In particular Moral [26] uses local propagation algorithm for the deduction process in classical propositional logic. Our approach can be viewed as an extension of their works where propositional formulas are associated with necessity degree. However, the first aim of this paper is strictly improving propagation algorithms in possibilistic networks, and not proposing an alternative implementation of classical or possibilistic logic.

Based on this approach, our future works will focus on searching for the possibility of using other operators such as product or maximum.

Acknowledgement

This work has been supported by the French national project MICRAC (Modeles informatiques et cognitifs du raisonnement causal).

References

[1] S.L. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, Journal of the Royal Statistical Society 50 (1988) 157–224.
[2] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmman, San Francisco (California), 1988.
[3] P. Fonck, Réseaux d'inférence pour le raisonnement possibiliste, Ph.D. Thesis, Université de Liège, Faculté des Sciences, 1994.
[4] F.V. Jensen, Introduction to Bayesien Networks, UCL Press, University college, London, 1996.
[5] S. Benferhat, S. Smaoui, Hybrid possibilistic networks, in: Proceeding of the Twentieth National Conference on Artificial Intelligence (AAAI-05), AAAI Press/The MIT Press, Pittsburgh, USA, 2005, pp. 584–589.
[6] S. Benferhat, S. Smaoui, Possibilistic networks with locally weighted knowledge bases, in: Proceedings of the Fourth International Symposium on Imprecise Probabilities and Their Applications (ISIPTA '05), Brightdocs, Pittsburgh, USA, 2005, pp. 41–50.
[7] D. Dubois, H. Prade, Possibility Theory: An Approach to Computerized, Processing of Uncertainty, Plenium Press, New York, 1988.
[8] E. Hisdal, Conditional possibilities independence and non-interaction, Fuzzy Sets and Systems 1 (1978) 283–297.
[9] B. Bouchon-Meunier, G. Coletti, C. Marsala, Independence and possibilistic conditioning, in: Annals of Mathematics and Artificial Intelligence, 2002, pp. 107–123.
[10] G. de Cooman, E.E. Kerre, A new approach to possibilistic independence, in: Proceedings of the Third IEEE International Conference on Fuzzy Systems, 1994, pp. 1446–1451.
[11] P. Fonck, A comparative study of possibilistic conditional independence and lack of interaction, International Journal of Approximate Reasoning 16 (1997) 149–171.
[12] G. de Cooman, Possibility theory III: possibilistic independence, International Journal of General Systems 25 (1997) 353–371.
[13] L. de Campos, J.F. Huete, Independence concepts in possibility theory: Part I, Fuzzy Sets and Systems 103 (1999) 127–152.
[14] N.B. Amor, S. Benferhat, Graphoid properties of qualitative possibilistic, independence, International Journal of Uncertainty, Fuzziness and Knowledge-Based 13 (2005) 59–96.
[15] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Information science 9 (1975) 43–80.
[16] D. Dubois, J. Lang, H. Prade, Possibilistic logicHandbook on Logic in Artificial Intelligence and Logic Programming, vol. 3, Oxford University press, 1994, pp. 439–513.
[17] J. Lang, Possibilistic logic: complexity and algorithms, in: Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 5, 2000, pp. 179–220.
[18] C. Borgelt, J. Gebhardt, R. Kruse, Possibilistic graphical models, in: Proceedings of International School for the Synthesis of Expert Knowledge (ISSEK'98), Udine (Italy), 1998, pp. 51–68.

[19] F. Lin, R. Reiter, Forget it! in: Proceeding of AAAI Fall Symposium on Relevance, 1994, pp. 154–159.
[20] J. Lang, P. Marquis, Complexity results for independence and definability, in: Proceeding of the 6th International Conference on Knowledge Representation and Reasoning (KR'98), 1998, pp. 356–367.
[21] A. Darwiche, P. Marquis, Compiling propositional weighted bases, Artificial Intelligence 157 (1–2) (2004) 81–113.
[22] S. Benferhat, D. Dubois, H. Prade, From semantic to syntactic approaches to information combination in possibilistic logic, Aggregation and Fusion of Imperfect Information, Studies in Fuzziness and Soft Computing (1997) 141–151.
[23] S. Benferhat, D. Dubois, L. Garcia, H. Prade, On the transformation between possibilistic logic bases and possibilistic causal networks, International Journal of Approximate Reasoning 29 (2) (2002) 135–173.
[24] N. Wilson, J. Mengin, Embedding logics in the local computation framework, Journal of Applied Non-Classical Logics 11 (3–4) (2001) 239–267.
[25] J. Mengin, N. Wilson, Logical deduction using the local computation framework, Lecture Notes in Computer Science (1999) 386–396.
[26] L. Hernandez, S. Moral, Inference with idempotent valuations, in: Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-97), Morgan Kaufmann, San Francisco, CA, 1997, pp. 229–237.